What is claimed:

1. A provisioning engine for provisioning communications services including at least one provisioning model for carrying out common provisioning work processes, said provisioning model defining a state machine comprising:

a set of plural current states of said provisioning model;

means for generating a signal;

at least one transition operative to define conditions under which states are added to or removed from said set of current states.

2. A provisioning engine as recited in claim 1, wherein said transition comprises a signal type, a set of from states, and a set of to states, and wherein said transition is operative to add said to states to said set of current states and remove said from states from said set of current states when a signal matching said signal type is received by an executing instance of said model and said from states are a subset of said set of current states.

3. A provisioning engine as recited in claim 2, wherein said transition is operative to modify arguments of the signal and return the signal to said means for generating a signal.

4. A provisioning engine as recited in claim 1, wherein said means for generating a signal comprises an external API of the provisioning engine.

5. A provisioning engine as recited in claim 1, wherein said means for generating a signal comprises one of said transitions.

6. A provisioning engine as recited in claim 1, wherein said means for generating a signal comprises means for generating a signal API call and means for delivering a signal at a predetermined time after the corresponding signal API call.

7. A provisioning engine as recited in a claim 1, wherein said model comprises plural executing instances each having means for storing data specific to the instance.

8. A provisioning engine as recited in claim 7, wherein said model comprises means for storing data to be used by each of said instances.

9. A provisioning engine as recited in claim 2, wherein said transition further comprises at least one task and means for executing said at least one task when a signal matching said signal type is received by an instance of said model and said from states are a subset of said current states.

10. A provisioning engine as recited in claim 9, wherein said transition comprises transition arguments that are communicated to said at least one task executed by said means for executing.

11. A provisioning engine as recited in claim 7 wherein at least one of the said instances includes calls to another model as a subinstance.

12. A provisioning engine as recited in claim 11, wherein said subinstance comprises means for communicating with said at least one instance.

13. A provisioning engine as recited in claim 12, wherein transitions of said at lease one instance are configured to stop executing while said subinstance executes and continue executing when said subinstance is done executing.

14. A pattern language for programming engines for provisioning communications services including a provisioning model defining a state machine for carrying out common provisioning operations, said pattern language including a plurality of objects representing state machine functions, each of said objects comprising:

a set of plural current states of said provisioning model;

at least one transition operative to define conditions under which states are added to or removed form said set of current states.

15. A pattern language as recited in claim 14, wherein at least one of said objects is a subinstance object which comprises calls to another provisioning model as a subinstance.

16. A pattern language as recited in claim 15, wherein said subinstance object comprises means for communicating with said subinstance.

17. A pattern language as recited in claim 16, wherein said calls comprise a URL designating an id of said subinstance.

18. A pattern language as recited in claim 17, wherein said calls further comprise arguments.

19. A pattern language as recited in claim 18, wherein said means for communicating comprises means for changing said arguments.

20. A pattern language as recited in claim 11, wherein said transition comprises a signal type, a set of from states, and a set of to states, and wherein said transition is operative to add said to states to said set of current states and remove said from states from said set of current states when a signal matching said signal type is received by an executing instance of said model and said from states are a subset of said set of current states.

21. A pattern language as recited in claim 2, wherein said transition is operative to modify arguments of the signal and return the signal to a source of signal origination.

22. A pattern language as recited in claim 13, wherein said source of origination of the signal is an external API of the provisioning engine.

23. A pattern language as recited in claim 13, wherein said source of origination of the signal is one of said transitions.

24. A pattern language as recited in claim 11, further comprising means for generating a signal API call and means for delivering a signal at a predetermined time after the corresponding signal API call.

25. A pattern language as recited in claim 12, wherein said transition further comprises at least one task and means for executing said at least one task when a signal matching said signal type is received by an executing instance of the model and said from states are a subset of said current states.

26. A pattern language as recited in claim 17, wherein said transition further comprises transition arguments that are communicated to said at least one task executed by said means for executing.